Subject: Masks in 3D with IDLanROIGroup
Posted by kyp4 on Mon, 10 Apr 2006 18:16:38 GMT
View Forum Message <> Reply to Message

I have searched around this group for this problem and, while similar issues have been discussed, I cannot seem to find my exact problem.

Here is what I am attempting to do: I have an IDLanROIGroup object (actually an IDLgrROIGroup object but that SHOULDN'T make a difference) that contains a 3D ROI in the format of a 2D ROI (IDLgrROI object) for each slice of a 3D volume. Therefore the vertices for any given IDLgrROI object in the group will all have the same z value.

For a given (x,y,z) point within the 3D ROI I am trying to generate a slice ROI contour in all three planes, using IDLgrROIGroup->ComputeMask and then CONTOUR to generate an ROI based on the mask. It works fine for the XY (actually for this I simply use the ROI in the group with the z value and therefore need not generate a mask first) and YZ planes but ComputeMask crashes when called for the XZ plane. More accurately I think the mask array returned is corrupt and so the crash occurs when trying to access it. I believe I am correctly using ComputeMask and so cannot understand why it is crashing:

```
; Get ROI bounds
oROIGroup->GetProperty, ROIGROUP_XRANGE=xr, ROIGROUP_YRANGE=yr,
ROIGROUP_ZRANGE=zr

; XZ plane
ms = [xr[1]-xr[0]+1,zr[1]-zr[0]+1]
mask = oROIGroup->ComputeMask(DIMENSIONS=ms, MASK_RULE=2,
LOCATION=[xr[0],xyz[1],zr[0]], PLANE_NORMAL=[0,1,0],
PLANE_XAXIS=[1,0,0])

; YZ plane
ms = [yr[1]-yr[0]+1,zr[1]-zr[0]+1]
mask = oROIGroup->ComputeMask(DIMENSIONS=ms, MASK_RULE=2,
LOCATION=[xyz[0],yr[0],zr[0]], PLANE_NORMAL=[1,0,0],
PLANE_XAXIS=[0,1,0])
```

I have tried isolating the calls (doing only XZ or only YZ) and the results are the same: XZ mask array is corrupt while the YZ is fine. I have also tried things like removing the dimensions (resulting in the default 100 by 100) but again the same thing happens.

On a somewhat unrelated note (or perhaps it is relevant) the mask returned in the YZ case, while correct, seems to be to be in a sort of unexpected, inverted format and I have to fix it like such:

```
mask = Reform(mask, ms[1], ms[0], /OVERWRITE)
mask = Rotate(Temporary(mask), 3)
```

Is this a bug in IDL or am I using ComputeMask incorrectly? I have searched the net for this problem before and I recall reading somewhere that it was a bug which was fixed in version 6.2, though I am unable to locate this information again. I have since upgraded from IDL 6.1 to 6.2 and it still does not seem to be working. Any help would be much appreciated.

-Dan Whitman (Kyp4)
